Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.047058

ARTICLE

# A Blockchain-Based Access Control Scheme for Reputation Value Attributes of the Internet of Things

**Hongliang Tian and Junyuan Tian***

College of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China

*Corresponding Author: Junyuan Tian. Email: dr.thl@mail.neepu.edu.cn

Received: 23 October 2023 Accepted: 27 November 2023 Published: 30 January 2024

**ABSTRACT**

The Internet of Things (IoT) access control mechanism may encounter security issues such as single point of failure and data tampering. To address these issues, a blockchain-based IoT reputation value attribute access control scheme is proposed. Firstly, writing the reputation value as an attribute into the access control policy, and then deploying the access control policy in the smart contract of the blockchain system can enable the system to provide more fine-grained access control; Secondly, storing a large amount of resources from the Internet of Things in Inter Planetary File System (IPFS) to improve system throughput; Finally, map resource access operations to qualification tokens to improve the performance of the access control system. Complete simulation experiments based on the Hyperledger Fabric platform. From the simulation experimental results, it can be seen that the access control system can achieve more fine-grained and dynamic access control while maintaining high throughput and low time delay, providing sufficient reliability and security for access control of IoT devices.

**KEYWORDS**

Blockchain; IoT; access control; Hyperledger Fabric

## 1 Introduction

The data, files, controllable hardware devices, and accessible programs present in the IoT [1] can be called resources. Most of these resources are only allowed to be accessed by authorized users. If users' resource access is not managed, it is easy to leak user information due to arbitrary access [2]. Reasonable and effective access control on these resources is one of the important means to protect users' privacy, but the disadvantage of the traditional access control model [3] is that a central entity is needed to manage the information, the problem with this approach is that the central entity is not completely trustworthy, and there is a risk of leakage; in addition, a single central entity is easy to be attacked, and if the central entity is breached, it also causes incalculable damage to users.

The consensus mechanism within the Blockchain Network [4] (BCN) provides assistance in creating a distributed platform with a high level of trust, which can register and manage users and resources in IoT, as well as authenticate and grant access permissions in an extremely low trust scenario, and under the protection of the SHA256 algorithm [5], the data information in the BCN is difficult to tamper with, and the data information stored in the BCN has traceability [6], which solves

the single point of failure and other security issues that exist in traditional centralized access control systems. At present, many scholars have combined BCN with access control to solve the problems of single point of failure and data information security, although good results have been achieved, the complex way of changing permissions makes it difficult for the system to achieve fine-grained, dynamic control; and it did not fully utilize the computing power of Smart Contract (SC) [7] itself; also because of the shortcomings of BCN itself leads to low throughput of the system.

To address the above issues, a blockchain-based Reputation and Token ABAC (RT-ABAC) scheme for IoT reputation attribute access control is proposed. Based on the traditional ABAC (Attribute Access Control) scheme, Hyperledger Fabric is selected as the blockchain network architecture, with blockchain serving as a barrier against external network attacks to improve the security of IoT devices. By using smart contracts for logical judgment of access control, the access control strategy is deployed in the form of SC in the BCN, and resource-related information is stored in the IPFS [8]. Based on fully utilizing the computing power of SC, the security of resource storage and system throughput are improved; Design a reputation value analysis mechanism, using Reputation Value (RV) as one of the attributes, and dividing $RV$ into different intervals. The size of the $RV$ will determine the permissions that RDs can apply for qualification tokens to ensure finer-grained access control [9]. A qualification token request mechanism has also been designed, introducing the concept of qualification tokens into access control policies. Qualification tokens serve as credentials to obtain access permissions in the system, and Resource Demanders (RDs) obtain access permissions by applying for qualification tokens in advance, as well as the method of applying for multiple resource permissions at once, to reduce the impact of BCN on access control performance.

## 2 Related Works

In the research on the combination of Ethereum and access control, Yang et al. [10–12] used SC to complete access control calculation and permission granting, while Zhai et al. [13–15] used SC to record access control information. Zhang et al. [16] and Alshehri et al. [17] combined Hyperledger Fabric with access control and write the access control policies into the SC to achieve dynamic access control effects. Literature based on incorporating access control policies into SC, Cui et al. [18] and Sultana et al. [19] also introduced the concept of tokens in the entire access control process, which can simplify the difficulty of access control by applying tokens. To increase the throughput of BCN, Muhammad [20] and Nizamuddin et al. [21] chose to upload the resources into IPFS first and store the hash values of the resources only in BCN. Liu et al. [22] proposed a blockchain-based big data access control mechanism based on the ABAC model, which manages access control policies and attributes information through blockchain transactions to ensure the immutability, auditability, and verifiability of access control information. However, this research work is still in the theoretical research stage and lacks specific implementation. Ding et al. [23] used blockchain technology to improve the access management of IoT devices, record the distribution of attributes using blockchain technology to avoid single points of failure and data tampering and optimize access control processes to meet the needs of efficient and lightweight computing for IoT devices. Gupta et al. [24] proposed a formal access control system based on attributes, introducing the concept of groups in the access control model. Different groups are assigned to users based on their attributes, and the author also sets access policies within the access control system based on their personalized privacy. Behrad et al. [25] considered the pressure brought by connecting providers during the authentication process of IoT devices, the author proposes a dedicated authentication and access control mechanism, which mainly utilizes the flexibility of virtual technology to allow the authentication of IoT devices to be entrusted to the providers of these devices, thereby reducing the device authentication pressure of connection providers.

Liu et al. [26], Zhang [27] and Liu [28] all addressed the issue of unclear permission allocation in the current access control model of the Internet of Things, which makes it easy for requesters to have unauthorized access when using device permissions, and the access control model cannot be applied to lightweight IoT devices. By combining blockchain technology with attribute-based access control models, a blockchain-based IoT access control model was proposed to improve fine-grained access control permissions through the coordination of multiple contracts.

Good results have been achieved in these studies mentioned above, but there is still the problem of too coarse granularity in the division of access control privileges, which can easily lead to security problems such as over-authorized access and over-authorization. Different from the above research, the RT-ABAC scheme proposed in this paper introduces a new attribute-reputation value, which can divide the access permissions of RDs and achieve more fine-grained permission granting. At the same time, unlike traditional token request mechanisms, this scheme allows RDs to make requests to multiple resources. If the user meets the access requirements of multiple resources at the same time, all resource IDs that meet the access requirements will be mapped to the qualification token. This not only achieves batch requests for resources by RDs, but also eliminates duplicate request operations by RDs, improves access efficiency, and does not generate a large number of qualification tokens, reducing the management pressure on BCNs.

## 3 Proposed Framework

The architecture of the blockchain-based Internet of Things reputation value attribute access control system is divided into three parts: user end, network end, and device end, as shown in Fig. 1. The user end includes two types of users: Resource Owners (ROs) and RDs. After completing user registration, ROs are responsible for registering resources on the network end and verifying the authenticity of the qualification tokens sent by RDs. After completing user registration, RDs will interact with the network end to apply for qualification tokens, and then carry the qualification tokens to apply for resources from ROs. The network side includes IPFS storage systems, blockchain, and internal channel deployment chaincode. Utilizing the IPFS system to store the raw data of resources, the IPFS system achieves secure storage of data through distributed hash table technology and stores the resource hash strings returned by the IPFS system in the BCN to reduce the storage pressure on the BCN. Hyperledger Fabric is selected as the blockchain network architecture, and chain contracts include User Register Smart Contract (URSC), Resource Register Smart Contract (RRSC), Token Application Smart Contract (TASC), and Token Validate Smart Contract (TVSC) enable the connection between users and devices and the storage of resource data through smart contracts. The device end of the system is composed of various IoT terminal devices, which are connected to the IoT themselves. Each device has a unique device ID to distinguish it from other devices. The device receives permission instructions transmitted by the system and executes commands issued by the user.

### *3.1 System Working Sequence*

The working sequence diagram of the access control system is shown in Fig. 2. The user first calls the smart contract for user registration, and after successful registration, the blockchain network will return the user's attribute weights. Subsequently, ROs store their resources in IPFS based on symmetric encryption. IPFS returns a hash string of encrypted data to ROs, and then they call the contract to register the resources. When RDs want to request permissions for one or more resources, they first apply for a qualification token by calling TASC and then send the qualification token returned by TASC to ROs for verification. After verification, RDs will receive the hash string and encryption key

k (key) sent by ROs. RDs use the hash string to retrieve the resources they need in IPFS, and decrypt k to obtain the original resources.

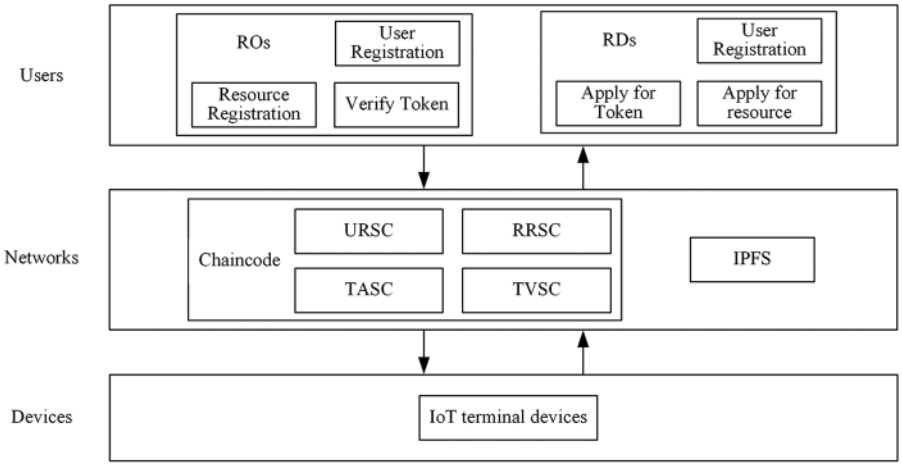


**Figure 1:** Access control system architecture diagram

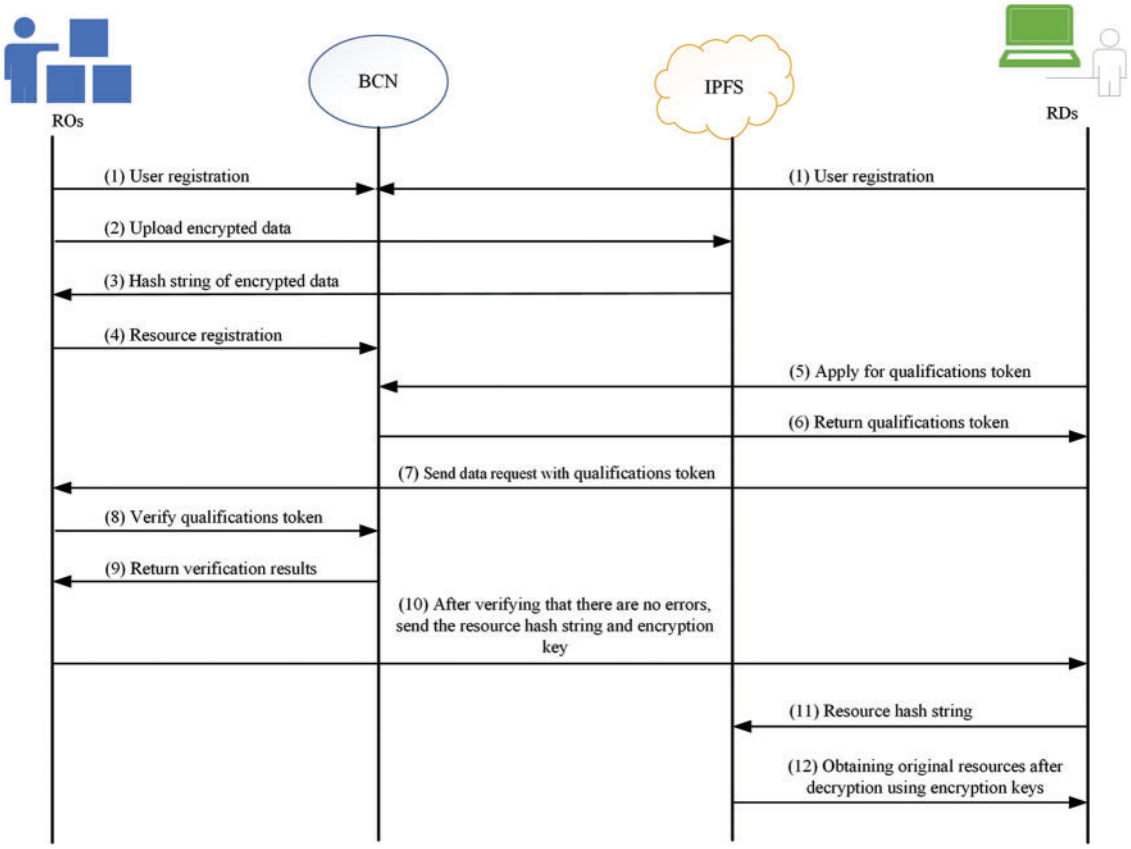


**Figure 2:** Access control system working sequence diagram

### *3.2 Registration Module*

After the user joins the system, the registration module calls the URSC deployed in the BCN to upload their account address $User_{Address}$ and personal information $User_{Information}$ to obtain the user attribute set $User_{AttrSet}$. The smart contract then calculates the user's attribute weight $User_{AttrPower}$ based on it. Algorithm 1 provides a brief description of the user registration process. When registering, users should ensure that the user address is not a duplicate registration to avoid the problem of duplicate registration of a user address. A status value of 1 indicates successful registration, otherwise, this value remains at 0. Algorithm 1 describes user registration on the chain.

**Algorithm 1:** User Registration

**Input:** $User_{Address}$, $User_{Information}$, $User_{AttrSet}$
**Output:** $User_{AttrPower}$
1.**function** *Register_User(* $User_{Address}$, $User_{AttrSet}$, $User_{Information}$ *)*
2.**if** $User_{Address} == 1$ *and* $User_{Information} == 1$ **then**
3. get = $User_{AttrSet}$
4.**else**
5. $User_{Address} == 1$ *and* $User_{Information} == 1$
6.**end if**
7.**for** $i = 0$ *to* $User_{AttrSet}.length\text{-}1$ **do**
8. $User_{AttrPower} = User_{AttrSet[i].power}$
9. **return** $User_{AttrPower}$
10.**end for**
11.**end function**

After the user successfully registers, they will register their resources, call the RRSC to upload the resource number $Res_{ID\{...\}}$, and encrypt it to generate a resource hash identification index $Res_{Hash}$. Then, the ROs will set the resource weight limit value $Res_{[P]PowerLimit}$ and different operation permissions P (permission) based on the reputation value size. Once the resource registration is successful, ROs will receive a resource status notification returned by RRSC, which means that the resource registration process has been completed. This resource registration method allows RDs to directly search for resource requests based on $Res_{ID\{...\}}$, making the request process for RDs more convenient. Algorithm 2 provides a description of resource registration on the chain.

**Algorithm 2:** Resource Registration

**Input:** $Res_{ID\{...\}}$, $Res_{Hash}$, $P$
**Output:** $Res_{[P]PowerLimit}$
1.**function** *Register_Resource(* $Res_{ID\{...\}}$, $Res_{Hash}$, $P$ *)*
2.**if** $Res_{ID\{...\}} == 1$ **then**
3. $Res_{Hash} = Hash(Res_{ID\{...\}})$
4. $Res[P]_{PowerLimit} = Res_{ID\{...\}}[P]$
5.**else**
6. $Res_{ID\{...\}} == 1$
7. **return** $Res_{[P]PowerLimit}$
8.**end if**
9.**end function**

### 3.3 Credit Value Analysis Module

The design of $RV$ is to constrain user bad behavior, identify malicious nodes, and implement trust constraints. BCN calculates reputation value attributes based on user history behavior records, and user and resource information it owns. The user's $RV$ is generally represented by any real number on [0, 1]. Assuming that there are only a small number of malicious users in the entire BCN. The calculation formula (1) for $RV_t$ is as follows:

$$RV_t = \sum_{SH} P(\mathrm{sh}_i) \times F(s, o, sh_i) \tag{1}$$

The formula utilizes the results of the user's previous sessions to compute the resource demander's $RV_t$ at the current time $t$, where $P(\mathrm{sh}_i) = p(\mathrm{sh}_i)/\sum p(\mathrm{sh}_j)$, denotes the share of session $sh$ in the session history set $SH$, and $p(\mathrm{sh}_i)$ decreases with time. It is assumed that each session behavior brings certain benefits, the electronic transaction brings actual benefits, and the sharing of resources and privilege management, etc. bring behavioral benefits. $F(s, o, sh_i) = f(X(sh) - X(\phi))$, denotes the trust assessment of the session $sh_i$, where $X(\phi)$ is the expected benefit,$X(sh)$ is the actual benefit and $f(x) = \sin(\pi x/2)$. According to the social behavioral science of trust behavior. The default $RV$ for users joining the system for the first time is randomly obtained between [0.1, 0.5], under similar conditions, the user takes the same behavior as a binomial event with approximate probability P. The closer to the expected probability, the higher the probability of occurrence, i.e., the closer the $RV$ of a session is to 1 when the gap between the expected and actual benefits is smaller. On the contrary, the higher the gap between the expected and actual benefits is, the lower the $RV$ generated by the session.

To make the granularity of the division of access control privileges more detailed, the method adopted in this paper is to divide the $RV$ into four different intervals, and only users whose $RV$ meets the standard can apply for the privileges corresponding to that interval. In the actual interaction process of accessing resources, it is assumed that most of the user behaviors are relatively trustworthy, and malicious users and advanced users are only a small part of them. Therefore, according to the above assumptions, the four $RV$ level intervals are set as [0, 0.1], (0.1, 0.5], (0.5, 0.9], and (0.9, 1) according to the normal distribution. The correspondence between $RV$ and access rights is shown in Table 1.

**Table 1:** Classification of user reputation value interval

| Privilege level | Credit value range | User level | Access authority |
|---|---|---|---|
| 1 | [0, 0.1] | Malicious user | Denial of access |
| 2 | (0.1, 0.5] | Power user | View resources |
| 3 | (0.5, 0.9] | Intermediate user | View, download resources |
| 4 | (0.9, 1) | Power user | View, download, and modify resources |

### 3.4 Qualification Token Application Module

When RDs want to access a certain resource or resources, they can apply for the $Res_{ID\{...\}}$ to specify a set of objects. After RDs send a request to the qualification token application module, TASC deployed in the BCN will be automatically triggered. TASC will interact with the access control policy in RRSC to verify whether the $User_{AttrPower}$ of RDs meet the requirements set by ROs. TASC will map the resource permissions that meet the requirements to the qualification token. Subsequently, BCN will interact

with ROs, inform them of the access request information, and finally return the access result and qualification token to RDs. Each qualification token has a number $Token_{ID}$, which is required for qualification tokens to be referenced in future command execution, further improving request speed. Algorithm 3 describes the process of RDs initiating qualification token requests based on $Res_{ID\{...\}}$.

**Algorithm 3:** Apply for Qualification Token

**Input:** $Res_{ID\{\dots\}}$, $P$, $User_{AttrPower}$, $Res_{[P]PowerLimit}$
**Output:** $Token_{ID}$
1.**function** $Request(Res_{ID\{\dots\}}, P, User_{AttrPower}, Res_{[P]PowerLimit})$
2.**if** $Res_{ID\{\dots\}} == 1$ **then**
3. **if** $User_{AttrPower} >= Res_{[P]PowerLimit}$ **then**
4. $Token_{ID} == Token_{[p]}$
5. $Token_{ID\rightarrow}$ $RDs$
6. **else**
7. $User_{AttrPower} >= Res_{[P]PowerLimit}$
8. **end if**
9.**else**
10. $Res_{ID\{\dots\}} == 1$
11. **return** $Token_{ID}$
12.**end if**
13.**end function**

### *3.5 Qualification Token Validation Module*

After obtaining the qualification $Token_{ID}$, RDs will send a request to ROs containing $Token_{ID}$ and $Res_{ID\{...\}}$. ROs can call TVSC to verify the $Token_{ID}$, and TVSC will verify the validity of the qualification token based on $Res_{ID\{...\}}$ and the $Res_{[P]PowerLimit}$ set by ROs. Only when the $User_{AttrPower}$ is greater than the $Res_{[P]PowerLimit}$ set by ROs for the resources they own can all qualification tokens of RDs be proven to be valid. After the qualification token passes the validity verification, the $Res_{ID\{...\}}$ and k will be returned to RDs. After obtaining the $Res_{ID\{...\}}$ and k, the resource's hash string can be used to retrieve and retrieve the encrypted resource from IPFS, and then k can be used to decrypt the encrypted resource, ultimately obtaining the original resource data. Algorithm 4 provides a simple description of the validation of qualification tokens.

**Algorithm 4:** Verify Qualification Token

**Input:** $Token_{ID}$, $Res_{ID\{\dots\}}$
**Output:** $Res_{Hash}$, $k$
1.**function** $verify(Token_{ID}, Res_{ID\{\dots\}})$
2.**if** $Res_{ID\{\dots\}} == 1$ **then**
3. $TVSC \leftrightarrow RRSC(Res_{[P]PowerLimit})$
4. **if** $Token_{ID} == 1$ **then**
5. **return** $Res_{Hash}$, $k$
6. **else**
7. $Token_{ID} == 1$
8. **end if**
9.**else**

(Continued)

**Algorithm 4 (continued)**

10. $Res_{ID\{\ldots\}} == 1$
11. **end if**
12. **end function**

## 4 Safety Analysis and Experimental Results

### *4.1 Safety Analysis*

In this section, we will comprehensively demonstrate the security of the proposed solution from three aspects: registration module, qualification token application module, and qualification token verification module. The results indicate that this scheme can meet the security requirements of IoT access control.

1) Security of registration module

Uplink operations on user identity or resource information can be considered as a registration event. During the registration phase, timestamp and $Res_{ID}$ sequence number are two important parameters for handling replay attacks. Timestamps can record the time of data creation, making it difficult for attackers to change the temporal order of data blocks. $Res_{ID}$ are used to track and identify data blocks. When a certain data block is tampered with, it will not be possible to generate the correct $Res_{ID}$, making the attack easier to detect. In addition, the hash function SHA-256 has the characteristics of strong input sensitivity and collision resistance, which improves the fault tolerance of the system.

2) Security of qualification token application module

When RDs apply for qualification tokens, they will perform interactive verification with the access control policy set in the TASC contract. Only when RDs reach the set $RV$ and $User_{AttrPower}$ is greater than $Res_{[P]PowerLimit}$ will they be allowed to access resources. The default $RV$ for RDs added to the system for the first time is randomly obtained between (0.1, 0.5]. According to the formula (1), the performance of RDs in the system will be calculated, and their $RV$ will be updated accordingly. When RDs actively participate in resource management and sharing behaviors, sh will exhibit a good state, resulting in a higher $RV$. On the contrary, it will result in a lower $RV$. The system uses this to determine whether RDs are allowed to participate in the resource access process, to reject malicious RDs from joining.

ROs will set $Res_{[P]PowerLimit}$ based on the $RV$ owned by RDs to further filter RDs of different $RV$ levels, achieving more fine-grained access control for users. The TASC contract assigns a minimum permission to each RDs to ensure that they can only access the minimum permissions required to complete the task, and even if a security event occurs, attackers can only access authorized resources. In addition, the access behavior of RDs will be fully monitored and recorded by the blockchain in the form of a "log", and this "log" cannot be tampered with, providing non-repudiation for the entire system.

3) Security of qualification token verification module

RDs submit qualification tokens for permission verification. The $Token_{ID}$ sequence numbers are sequential and unique, ensuring the correctness of qualification tokens when accessing resources. RDs and ROs verify and confirm the validity of qualification tokens through blockchain-distributed consensus algorithms. If there is forgery or tampering with qualification tokens on the RDs side, it will conflict with the original correct $Token_{ID}$ and cannot pass the validity verification on the ROs side.

In summary, this solution can resist common risk attacks in IoT and ensure the security of the entire framework.

### *4.2 Experimental Results*

This section mainly tests the performance of the system by simulating the key steps in each module. References [26–28] all designed and implemented an IoT access control system based on the Hyperledger Fabric platform, which is combined with BCN. The purpose is to ensure the security and reliability of the access control mechanism in IoT and the access control permissions of each user. Similar to the design concept of this article, the access control policy is written into the SC, and the SC determines whether RDs are qualified to obtain certain permission for a certain resource, Therefore, references [26–28] were selected as the comparative literature.

To demonstrate the availability of the proposed scheme, we deployed the model of the scheme in the 17th version of VMware virtual machine, with a system configuration of dual-core 2 GB memory and 100 GB hard disk with Ubuntu 22.04.2 LTS 64bit, each module runs in the Docker container. Use the Hyperledger Caliper deployed in the Hyperledger Fabric as a performance testing tool. The software configuration of the testing environment is shown in Table 2, and the hardware configuration is shown in Table 3. The testing method is to initiate concurrent access requests to the Hyperledger Fabric, with 100, 200, 300, 400, and 500 requests, respectively, with write and read request types. The Workers in the experimental system are 5, the txNumber is 5000, and the txDuration is 30 s. Compare the performance of resource writing and reading between the two schemes under different concurrent access times.

**Table 2:** Software configuration table

| Software environment | Detailed information |
|---|---|
| Operating system | Ubuntu 22.04.2 LTS |
| Docker | v 20.10.25 |
| Docker-Compose | v 1.24.0 |
| Golang | v 1.17.1 |
| Hyperledger fabric | v 2.5.4 |
| Hyperledger caliper | v 0.4.2 |
| Node | v 16.20.2 |
| Npm | v 8.19.4 |

**Table 3:** Hardware configuration table

| Hardware environment | Detailed information |
|---|---|
| CPU | i5-6300HQ |
| Memory | 8 GB |
| Hard disk | 100 GB |

1) Experimental results of resource write throughput and average time delay

The comparison of resource write throughput is shown in Fig. 3. From the obtained data comparison graph, it can be seen that the resource write throughput of this scheme is greater than the resource write throughput in reference [26] and reference [27]. In comparison with reference [28], the resource write throughput of this scheme is slightly lower in the early stage, due to the side chain structure used for concurrent access operations in reference [28] and not attached to the main chain. But after the number of concurrent accesses exceeds 300, the throughput of this scheme will exceed that of reference [28]. This is because as the number of concurrent accesses increases, the data on the main chain in reference [28] will expand, leading to network congestion during main-side chain interaction. When storing resources, this scheme first uploads them to IPFS. This process will partition the resources, and the partitioned resources will be encrypted separately and mapped by IPNS to generate IPNS names. Parsing the IPNS name can obtain fragments of the resource hash string, and connecting multiple fragments can obtain the complete hash string of the resource. Due to the introduction of a sharding mechanism for resource writing, the throughput of resource writing is higher than that in the comparative literature. When the number of concurrent accesses is 500, the resource write throughput of this scheme reaches its maximum of 85TPS.

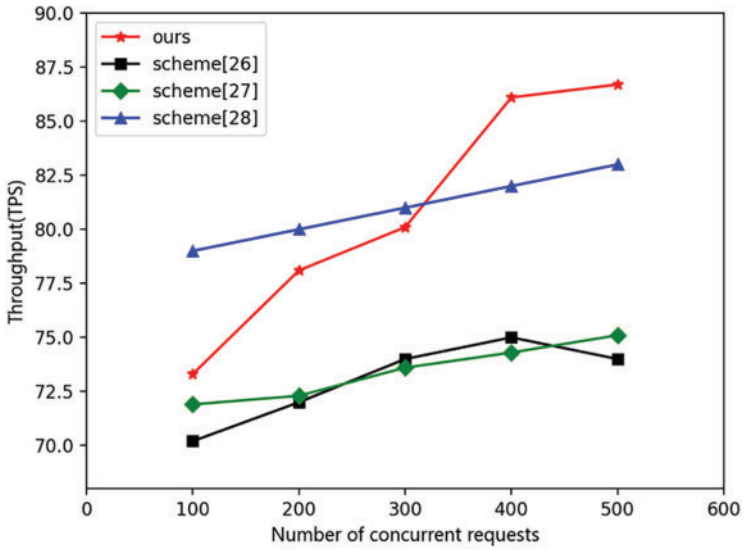


**Figure 3:** Resource write throughput comparison

The average time delay comparison of resource writes is shown in Fig. 4. In 500 concurrent access operations, the average time delay of resource writes in this scheme is around 0.25–0.3 s, and the average time delay of resource writes is lower than the average time delay of resource writes in the comparative literature. This is because the relevant attribute information in the access control strategy of this scheme is allocated by the chain code contract based on the user's information, while the relevant attribute information in the access control strategy of the comparative literature is the IP address and MAC address of the resource that requires time to interact with the IoT device, resulting in fluctuations in the average time delay of resource writing.

2) Experimental results of resource read throughput and average time delay

From Fig. 5, it can be seen that the resource read-throughput of this scheme is superior to that of the comparison scheme. In 500 concurrent access operations, the resource read throughput of this

scheme remains above 250TPS, with a maximum of 292TPS. Compared to the reference, the maximum resource read throughput is only 240TPS. This is because qualification tokens have been introduced in this scheme, allowing RDs to initiate access requests to multiple resources simultaneously, greatly improving system performance. Fig. 5 shows that the throughput of both parties was initially similar, due to the initial process of applying for qualification tokens in RDs. However, as the number of visits increased, the throughput in this scheme showed a gradual upward trend, and the throughput in references [26] and [27] tended to stabilize, while the throughput in references [28] showed a decreasing trend.

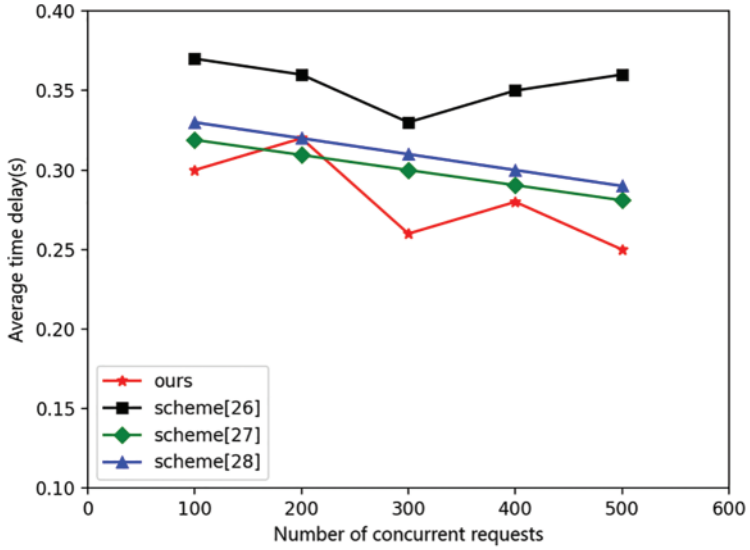


**Figure 4:** Resource write average time delay comparison

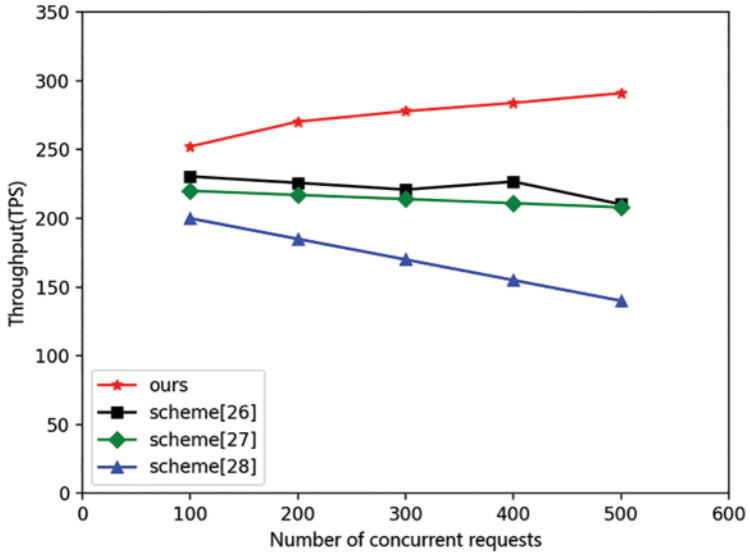


**Figure 5:** Comparison of throughput of resource reads

In the comparison of average time delay, because this scheme applies for qualification tokens in advance, it will save resource access time. From the data in Fig. 6, it can be seen that in 500 concurrent access operations, the average time delay of this scheme is always below 0.05 s, with a minimum of 0.03 s. Compared to the literature, the minimum average time delay is about 0.06 s.

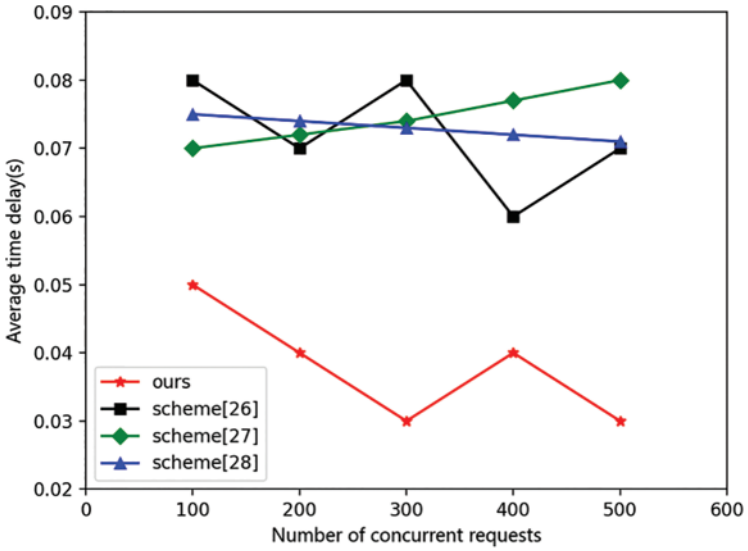


**Figure 6:** Comparison of the average time delay of resource reads

The above two sets of experiments demonstrate that this scheme has better performance compared to similar research schemes. The performance comparison table is shown in Table 4. It has good throughput and average time delay when writing and reading resources, and can withstand larger scale usage requests. This proves that introducing RV as an attribute in this scheme can refine access control policies and map resource access permissions to qualification tokens, It is a feasible solution to improve the throughput of the entire system and the security of access control policies by applying for access permissions to a certain or certain resources at once.

**Table 4:** Performance comparison table

| | Write throughput | Read throughput | Write average delay | Read average delay |
|---|---|---|---|---|
| Ours | Maximum | Maximum | Minimum | Minimum |
| Scheme [26] | Minimum | Middle | Maximum | Maximum |
| Scheme [27] | Middle | Middle | Middle | Middle |
| Scheme [28] | Middle | Minimum | Middle | Middle |

## 5 Conclusion

In this article, we introduce the concept of attributes in ABAC and consider RV as one of the attributes, proposing an RT-ABAC scheme combined with BCN. By refining attributes to reduce the difficulty of access control management, an open, transparent, secure, and trustworthy access control environment has been created. By utilizing IPFS storage resources, the performance of BCN

has been improved, and access control policies have been deployed to BCN through SC, achieving distributed access control effects and effectively preventing a single-point-of-failure issue. Introducing qualification tokens into the strategy, RDs obtain access permissions by applying for qualification tokens in advance and can apply for multiple resource access permissions at once, reducing the impact of BCN on access control performance and improving the throughput of the entire BCN. However, the RT-ABAC scheme proposed in this article lacks effective methods for encrypting user-side information. A more lightweight attribute-based encryption method can be designed to further reduce the resource write time delay required in the access control process.

**Acknowledgement:** The authors would like to express their gratitude to the members of the research group for their support.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Hongliang Tian; simulation, analysis, interpretation of results and draft manuscript preparation: Junyuan Tian. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Data not available due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] D. G. Li, R. Du, Y. Fu and M. H. Au, "Meta-key: A secure data-sharing protocol under blockchain-based decentralized storage architecture," *IEEE Networking Letters*, vol. 1, no. 1, pp. 30–33, 2019.

[2] N. Feng, Y. Chen and H. Feng, "Toout source or not: The impact of information leak age risk on information security strategy," *Information & Management*, vol. 57, no. 5, pp. 103215, 2020.

[3] S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "A role-based access control model in modbus SCADA systems. A centralized model approach," *Sensors*, vol. 19, no. 20, pp. 4455, 2019.

[4] P. Zhu, J. Hu, X. Li and Q. Zhu, "Using blockchain technology to enhance the traceability of original achievements," *IEEE Transactions on Engineering Management*, vol. 70, no. 5, pp. 1693–1707, 2023.

[5] D. Wu, Z. Xu and B. Chen, "Enforcing access control in information-centric edge networking," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 353–364, 2021.

[6] P. Zhu, J. Hu and X. Li, "Enhancing traceability of infectious diseases: A blockchain-based approach," *Information Processing & Management*, vol. 58, no. 4, pp. 102570, 2021.

[7] N. Nizamuddin, K. Salah and A. M. Azad, "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[8] Y. Hyoeun and P. Sejin, "Reliable vehicle data storage using blockchain and IPFS," *Electronics*, vol. 10, no. 10, pp. 1130–1145, 2021.

[9] L. Zhang, M. Peng and W. Wang, "Secure and efficient data storage and sharing scheme based on double blockchain," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 499–515, 2021.

[10] J. Yang, S. He, Y. Xu, L. Chen and J. Ren, "Atrustedroutingschemeusing blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, pp. 970, 2019.

[11] K. Košťál, P. Helebrangt and M. Belluš, "Management and monitoring of IoT devices using blockchain," *Sensors*, vol. 19, no. 4, pp. 856, 2019.

[12] J. Qiu, Z. Tian and C. Du, "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.
[13] P. Zhai, J. He and N. Zhu, "Blockchain-based Internet of Things access control technology in intelligent manufacturing," *Applied Sciences*, vol. 12, no. 7, pp. 3692, 2022.
[14] P. Zhu, J. Hu, X. Li, X. Jiang and M. X. Zhu, "The influences of livestreaming on online purchase intention: Examining platform characteristics and consumer psychology," *Industrial Management and Data Systems*, vol. 123, no. 3, pp. 8620–8885, 2022.
[15] A. R. Rajput, Q. Li and M. T. Ahvanooey, "EACMS: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84304–84317, 2019.
[16] Y. Zhang, S. Kasahara and Y. Shen, "Smart contract-based access control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.
[17] S. Alshehri and O. Bamasag, "AAC-IoT attribute access control scheme for IoT using lightweight cryptography and hyperledger fabric blockchain," *Applied Sciences*, vol. 12, pp. 8111, 2022.
[18] M. Cui, D. Han and J. Wang, "An ef cient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet Things*, vol. 6, no. 5, pp. 90769084, 2019.
[19] M. Sultana, A. Hossain and F. Laila, "Towards developing a secure medical image sharing system based on zero trust principles and blockchain technology," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 256, 2020.
[20] U. J. Muhammad, "Blockchain-based secure data storage for distributed vehicular networks," *Applied Sciences*, vol. 10, no. 6, pp. 2011, 2020.
[21] P. Nath, J. R. Mushahary and U. Roy, "AI and Blockchain-based source code vulnerability detection and prevention system for multiparty software development," *Computers and Electrical Engineering*, vol. 106, pp. 108607, 2023.
[22] A. D. Liu, X. H. Du and N. Wang, "A blockchain-based access control mechanism for big data," *Journal of Software*, vol. 30, no. 9, pp. 2636–2654, 2019.
[23] S. Ding, J. Cao and C. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.
[24] M. Gupta, F. M. Awaysheh and J. Benson, "An attribute-based access control for cloud-enabled industrial smart vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4288–4297, 2020.
[25] S. Behrad, E. Bertin and S. Tuffin, "A new scalable authentication and access control mechanism for 5G-based IoT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.
[26] H. Liu, D. Han and D. Li, "Fabric-IoT: A blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.
[27] J. H. Zhang, "Design and implementation of Internet of Things access control system based on blockchain," Ph.D. dissertation, Inner Mongolia University, China, 2021.
[28] J. L. Liu, "Research on distributed access control mechanism of Internet of Things based on blockchain," Ph.D. dissertation, Tianjin University of Techology, China, 2022.